

LABORATORY REPORT

Application Development Lab (CS33002)

B.Tech Program in CSE

Submitted By

Name:- Apratim Dutta

Roll No: 2205274



Kalinga Institute of Industrial Technology  
(Deemed to be University) Bhubaneswar,  
India

Spring 2024-2025

Table of Content

Exp No.	Title	Date of Experiment	Date of Submission	Remarks
1.	Build a Resume using HTML/CSS	16/01/2025	23/01/2025	
2.	Machine Learning for Cat and Dog Classification	23/01/2025	30/01/2025	
3.				
4.				
5.				
6.				
7.				
8.				
9.	Open Ended 1			
10.	Open Ended 2			
Lab Number		4		
Experiment Number		2		
Experiment Title		Machine Learning for Cat and Dog Classification		
Date of Experiment		23/01/2025		
Date of Submission		30/01/2025		

1. Objective:-

To classify images as cats or dogs using machine learning models.

2. Procedure:- (Steps Followed)

1. Collect a labeled dataset of cat and dog images.
2. Preprocess images using OpenCV (resize, flatten, etc.).

3. Train ML models: SVM, Random Forest, Logistic Regression, and K-means Clustering.
4. Save the trained models.
5. Build a Flask backend to load models and handle image uploads.
6. Create a frontend with HTML/CSS for uploading images and selecting models.
7. Display the classification result on the webpage.

### 3. Code:-

```
requirements.txt X app.py 2, U X index.html 1, U styles.css U scripts.js U
app > app.py > ...
3 import tensorflow as tf
4 from tensorflow.keras.preprocessing import image
5 from tensorflow.keras.applications.vgg16 import preprocess_input
6 import numpy as np
7 import pickle
8
9 app = Flask(__name__, static_folder='static')
10
11 # Load Models
12 try:
13     cnn_model = tf.keras.models.load_model('app/models/cnn_model.h5') # Replace with your CNN model file
14     with open("app/models/svm_model.pkl", "rb") as f:
15         svm_model = pickle.load(f)
16     with open("app/models/rf_model.pkl", "rb") as f:
17         rf_model = pickle.load(f)
18     with open("app/models/lr_model.pkl", "rb") as f:
19         lr_model = pickle.load(f)
20     with open("app/models/kmeans_model.pkl", "rb") as f:
21         kmeans_model = pickle.load(f)
22     vgg_model = tf.keras.applications.VGG16(weights='imagenet', include_top=False, input_shape=(150, 150, 3))
23 except Exception as e:
24     print(f"Error loading models: {e}")
25     exit()
26
27 IMG_HEIGHT = 150
28 IMG_WIDTH = 150
29
30 UPLOAD_FOLDER = 'app/uploads'
31 app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
32 os.makedirs(UPLOAD_FOLDER, exist_ok=True)
33
34 ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg', 'gif'}
35
36 Qodo Gen: Options | Test this function
37 def allowed_file(filename):
38     return '.' in filename and \
39         filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS
40
41 Qodo Gen: Options | Test this function
42 def prepare_image(img_path):
43     img = image.load_img(img_path, target_size=(IMG_HEIGHT, IMG_WIDTH))
```

```
requirements.txt X app.py 2, U index.html 1, U X styles.css U scripts.js U
app > templates > index.html > ...
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8" />
5 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6 <title>Cat-Dog Classifier</title>
7 <link rel="stylesheet" href="/static/styles.css" />
8 </head>
9 <body>
10 <div class="container">
11 <h1>Cat-Dog Classifier</h1>
12 <input type="file" id="fileInput" />
13 <button onclick="uploadImage()">Upload and Predict</button>
14 <div id="result"></div>
15 <img
16 id="uploadedImage"
17 src=""
18 alt="Uploaded Image"
19 style="display: none"
20 />
21 </div>
22 <script src="/static/scripts.js"></script>
23 </body>
24 </html>
25
```

```
AD Lab 4 Exp 2.ipynb Save failed
File Edit View Insert Runtime Tools Help
+ Code + Text Reconnect 14 Gemini

[ ] try:
    # This command only in Colab.
    %tensorflow_version 2.x
except Exception:
    pass
import tensorflow as tf

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, Flatten, Dropout, MaxPooling2D
from tensorflow.keras.preprocessing.image import ImageDataGenerator

import os
from numpy import round
import matplotlib.pyplot as plt

Colab only includes TensorFlow 2.x; %tensorflow_version has no effect.

[ ] # Get project files
!wget https://cdn.freecodecamp.org/project-data/cats-and-dogs/cats_and_dogs.zip

!unzip cats_and_dogs.zip

PATH = 'cats_and_dogs'

train_dir = os.path.join(PATH, 'train')
validation_dir = os.path.join(PATH, 'validation')
test_dir = os.path.join(PATH, 'test')

# Get number of files in each directory. the train and validation directories

# 3
train_image_generator = ImageDataGenerator(rescale = 1./255)
validation_image_generator = ImageDataGenerator(rescale = 1./255)
test_image_generator = ImageDataGenerator(rescale = 1./255)

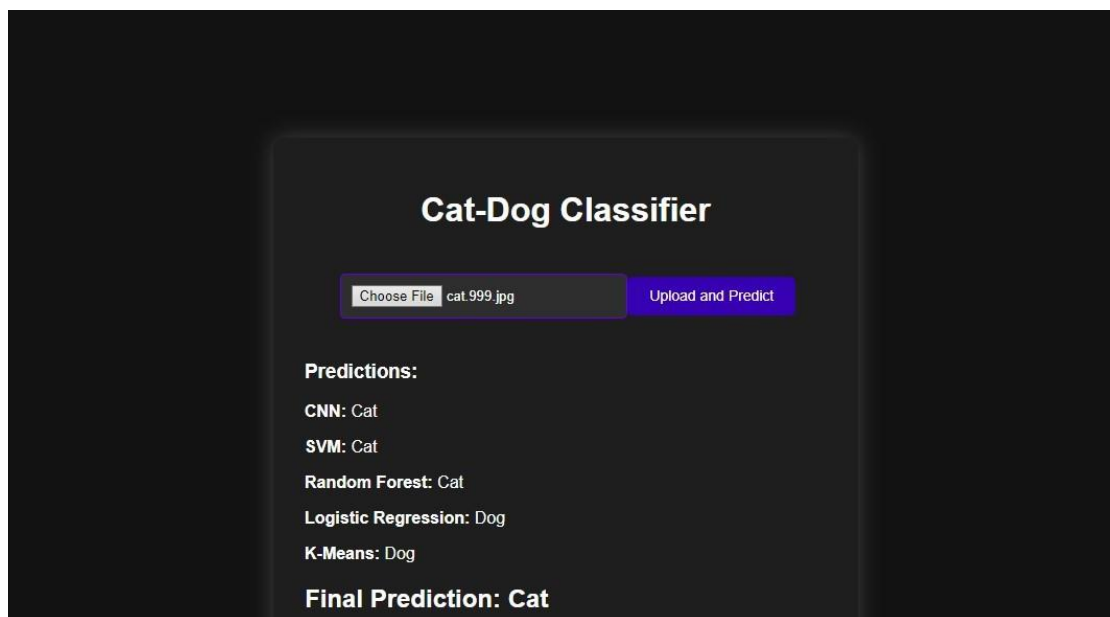
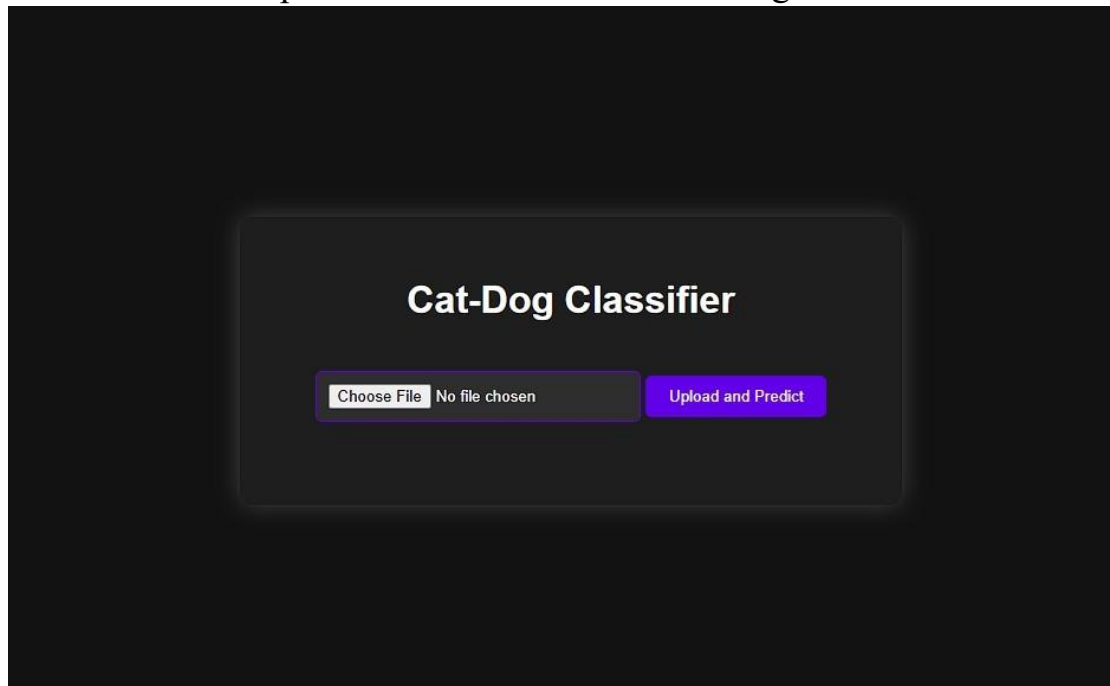
train_data_gen = train_image_generator.flow_from_directory(
    train_dir,
    batch_size=batch_size,
    target_size=(IMG_HEIGHT, IMG_WIDTH),
    class_mode='binary')

val_data_gen = validation_image_generator.flow_from_directory(
    validation_dir,
    batch_size=batch_size,
    target_size=(IMG_HEIGHT, IMG_WIDTH),
    class_mode='binary')

test_data_gen = test_image_generator.flow_from_directory(
    '/content/cats_and_dogs',
    classes=['test'],
    batch_size=batch_size,
    target_size=(IMG_HEIGHT, IMG_WIDTH),
    class_mode='binary',
    shuffle=False)

# 4
def plotImages(images_arr, probabilities = False):
    fig, axes = plt.subplots(len(images_arr), 1, figsize=(5, len(images_arr) * 3))
    if probabilities is False:
        for img, ax in zip( images_arr, axes):
            ax.imshow(img)
            ax.axis('off')
    else:
        for img, probability, ax in zip( images_arr, probabilities, axes):
            ax.imshow(img)
            ax.axis('off')
            if probability > 0.5:
                ax.set_title("%s %s" % (probability*100, "%"))
            else:
                ax.set_title("%s %s" % (probability*100, "%"))
```

4. Results/Output:- Entire Screen Shot including Date & Time



5. Remarks:-

\_\_\_\_\_  
(Name of the Coordinator)

Signature of the Student

\_\_\_\_\_  
(Name of the Student)

Signature of the Lab  
Coordinator